

METHOD FOR REPORTING TERMINAL INFORMATION, AND METHOD AND SYSTEM FOR  
MAINTAINING TERMINAL DEVICE

**Field of the Invention**

5 [0001] The invention relates to communication technology field, especially to a method and a system for maintaining terminal device and management.

**Background of the Invention**

10 [0002] Mobile terminals are important components of the entire mobile operation service system. As the functions of terminals become increasingly sophisticated, it is more likely that errors occur in terminal software. The competition among operators will be increasingly intense in the future; therefore, how to effectively  
15 maintain efficient Quality of Service (QoS) and low-cost device maintenance has become a major concern for operators and terminal manufacturers.

[0003] Device Management, as an important part in the Open Mobile Alliance (OMA) specifications, has specified some fundamental  
20 standard protocols for device management. However, it has not covered all aspects of device management. For example, with respect to the functionality of device fault resolution, in the existing OMA Device Management (DM) process, if an error occurs on a device, the user of the device has to notify the operator by some means (e.g., phone,  
25 WEB site, WAP site, etc.), then the operator's DM system initiates a resolution process to implement terminal configuration, as described hereunder:

[0004] 1. The user reports the error to the customer service center , for example, by means of a voice call;

30 [0005] 2. The customer service center queries the device for device

information;

[0006] 3. The device reports its relevant configuration information including device type, device serial number, Operating System (OS) version, capability, list of installed applications, event/performance logs, etc. to the customer service center;

[0007] 4. Based on this information, the person in the customer service center determines the causes of the error and requests the user for authorization to download the fault resolution software;

[0008] 5. The user grants the authorization;

[0009] 6. The customer service center pushes the application to the device; and the device installs and executes the application;

[0010] 7. The device sends an acknowledgement message to the customer service center.

[0011] The above method has the following drawbacks:

[0012] The above method requires the user to detect the error actively and notify the operator of the error before the fault resolution assistance may be provided. Such a method restricts the ways available for the operator to obtain errors in user terminals.

[0013] If the user has little knowledge of technical terms of the terminal, it is difficult for the user to report the terminal information accurately to the operator.

[0014] The terminal software developer has to handle software errors in a passive way; as a result, error detection and handling are often delayed.

### Summary of the Invention

[0015] In view of the defects and drawbacks in the prior art as described above, the present invention provides a method of reporting terminal information and a method and a system for maintaining terminal device in device management, so as to overcome the problems

in the prior art that there are few ways to obtain user terminal information, and it is difficult to obtain terminal information timely and accurately.

**[0016]** The method of reporting terminal information includes the following steps:

**[0017]** reporting, by a terminal device, the information of the terminal through a software program interface provided by a Device Management (DM) Agent module;

**[0018]** forwarding, by the DM Agent module, the terminal information to a DM Server; the DM Server reporting the terminal information to a Maintenance Unit (MU).

**[0019]** the software program interface includes a messaging interface, a file interface, an API, or a Web service interface.

**[0020]** the messaging interface includes an XML interface or a network protocol interface.

**[0021]** When the software program interface employs the API, the terminal information is combined into an XML format and is transmitted to the API as an argument.

**[0022]** The DM Agent module transmits the terminal information via an extended Open Mobile Alliance (OMA) DM protocol.

**[0023]** The transmission of the terminal information by the DM Agent module is implemented with a command of the extend OMA DM protocol which supports active event triggered by clients; or

**[0024]** the transmission of the terminal information by the DM Agent module is implemented by extending a standard command of the OMA DM protocol into a terminal information reporting command; or

**[0025]** the transmission of the terminal information by the DM Agent module is implemented by adding a special terminal information reporting command into the OMA DM protocol; or

**[0026]** the transmission of the terminal information by the DM Agent

module is implemented with a command of the OMA DM protocol directly.

**[0027]** The terminal information includes error information created during the operation of the terminal software, error information created by the terminal hardware, and process information created during the operation of the terminal.

**[0028]** The method for maintaining terminal device includes the following steps:

**[0029]** reporting, by the terminal device, the information of the terminal through a software program interface provided by a Device Management (DM) Agent module;

**[0030]** forwarding, by the DM Agent module, the terminal information to a DM Server;

**[0031]** reporting, by the DM Server, the terminal information to a Maintenance Unit (MU);

**[0032]** when receiving the terminal information, determining, by the MU, the corresponding software update package and sending the software update package to the DM Server;

**[0033]** maintaining, by the DM Server, the terminal device with the software update package following an OMA DM process.

**[0034]** upon receiving the terminal information, judging, by the DM Server, whether the terminal device may be maintained automatically; if the judgment is "Yes", maintaining, by the DM server, the terminal device directly following the OMA DM process; otherwise the method proceeds to the subsequent steps.

**[0035]** The software program interface includes a network protocol interface, an XML interface, or an API.

**[0036]** When the software program interface employs the API, the terminal device program will combine the terminal information into an XML format and send the terminal information to the API as an argument.

**[0037]** The DM Agent module transmits the terminal information via an extended OMA DM protocol.

**[0038]** The transmission of the terminal information by the DM Agent module is implemented with commands supporting active event triggered by clients in the extend OMA DM protocol; or

**[0039]** the transmission of the terminal information by the DM Agent module is implemented by extending a standard command of OMA DM protocol into a terminal information reporting command; or

**[0040]** the transmission of the terminal information by the DM Agent module is implemented by adding a special terminal information reporting command into the OMA DM protocol; or the transmission of the terminal information by the DM Agent module is implemented with a command of the OMA DM protocol directly.

**[0041]** The terminal information includes error information created during operation of the terminal software, error information created by terminal hardware, and process information created during operation of the terminal.

**[0042]** The Device Management (DM) system includes a DM Server adapted to manage a terminal device, a DM Agent module located in the terminal device and interacting with the DM Server, and a Maintenance Unit (MU) coupled to the DM Server and adapted to acquire, store, and maintain the information of the terminal device;

**[0043]** The DM Agent modules and the DM Server have a software program interface respectively;

**[0044]** The software program interface of the DM Agent module is adapted to receive the terminal information reported from the terminal devices and forward the terminal information to the DM Server; the DM Server reports the terminal information to the MU.

**[0045]** The software program interface is a messaging interface, a file interface, an API, or a Web service interface. With the software

program interfaces provided by the DM Agent module and the DM Server for information reporting, the terminal device software may report terminal information to the DM module, and enables the DM Agent module to automatically report the terminal information in an extended  
5 protocol to the DM Server, which in turn reports the terminal information to the MU, so that the terminal software developer may obtain the terminal information from the terminal software timely and accurately and thereby perform maintenance as appropriate. The invention may enhance the automatic fault resolution capability of  
10 the terminal software and simplify the DM fault resolution process, providing ease for user operation and management.

#### **Brief Description of the Drawings**

[0046] Fig.1 is a schematic of the DM system architecture according  
15 to an embodiment of the present invention;

[0047] Fig.2 is a flow chart of reporting terminal information from the terminal device according to an embodiment of the present invention;

[0048] Fig.3 is a flow chart of reporting terminal information from  
20 the terminal device and resolving the terminal error according to an embodiment of the present invention.

#### **Detailed Description of the Embodiments**

[0049] Referring to Fig.1, the DM system in an embodiment of the  
25 present invention includes DM Agent modules located in terminal devices, a DM Server, and a Maintenance Unit (MU).

[0050] The DM Agent module has a software program interface adapted for information reporting (hereinafter referred to as "information reporting interface"), which may be open to terminal Operation System  
30 (OS) and standard software on terminal OS as well as third-party

software developers. The application software and firmware/OS of the terminal device reports terminal information to the DM Agent module via the information reporting interface.

**[0051]** The DM Agent module interacts with the DM Server via an OMA  
5 DM protocol interface.

**[0052]** Similarly, there is a standard information reporting interface at the DM Server side. The DM Server reports the received terminal information to the MU via the information reporting interface; the MU may be a maintenance unit deployed by the terminal  
10 software developer.

**[0053]** The terminal information includes error information created by the software of the terminal device, error information created by the hardware of the terminal device, and process information created during the operation of the terminal device. The reported  
15 process information may be utilized as a condition for triggering automatic resolution in case of any error of the terminal software, and may be used to provide debugging information during software development process, acquire software use information during the trial period, acquire information in its beneficial use, and provide  
20 a mechanism to timely feedback to end user in case of any error or abnormality of the software. The terminals include mobile terminals or fixed terminals.

**[0054]** Hereunder the implementation of the invention will be described in detail, assuming the terminal information as software  
25 error information.

**[0055]** Terminal error information is described in a XML format, which may be applied in the information reporting interface and the reporting protocol according to the present invention. The following table describes the relevant description tags that may be used in  
30 the error information description format for constituting complete

error description information.

Tag	Description
Report	Container tag of terminal information, adapted to contain a piece of terminal information
Info	Brief description of error
Level	Error Class
AppName	Application name
Provider	Application provider
Address	Entry address of automatic information reporting interface of the application provider
Version	Version of application software
DataType	Type of error data
Data	Error data
Date (Time)	Time
OS	Operating system
OSVersion	Version of operating system
MemorySize	Memory size

**[0056]** The resulting XML format is as follows:

```
5      <Report>
      <Info></Info>
      <Level></Level>
      <AppName></AppName>
      <Provider></Provider>
10     <Address></Address>
      <Version></Version>
      <DataType></DataType>
      <Data></Data>
      <Date><Date>
15     <OS></OS>
      <OSVersion></OSVersion>
      <MemorySize></MemorySize>
      </Report>
```

**[0057]** The above-mentioned information reporting interface includes  
20 a messaging interface, a file interface (or the exchanging of data  
by means of file handle provided in OS), an Application Programming

Interface (API), or a Web service interface, etc, or any other appropriate interfaces. Wherein the messaging interface includes network protocol interface (e.g., TCP/IP interface, HTTP interface, UDP interface, or FTP interface) or an XML interface, etc.; the API is not limited to common local method invocation, but may also be used in remote method invocation by means of CORBA, Web Service (SOAP), RMI/IIOP, or DCOM, etc.

**[0058]** When the information reporting interface is API, the arguments of which may be in any of the following forms (only for reference, terminal information may be described in a more normalized and complete form when the standard is set down; there is no restriction in this aspect in the invention):

**[0059]** (1) error description format

**[0060]** The system provides a Notify API for terminal software to report errors; the API is similar to:

void Notify (char info[]); The argument is in an XML format that conforms to the error description format.

<Report>

<Info>unknown error</Info>

<Level>1</Level>

<AppName>TestApp</AppName>

<Provider>Huawei</Provider>

<Address>http://support.huawei.com/dmtest/Notify</Address>

<Version>1.0.1</Version>

<DataType>text/plain</DataType>

<Data>report reason</Data>

<Date>20050428163030</Date>

<OS>linux</OS>

<OSVersion>1.1.1</OSVersion>

<MemorySize>16M</MemorySize>

</Report>

**[0061]** The approach is favorable for extending corresponding

terminal error information; in the approach, the terminal software combines error information into the corresponding XML format.

**[0062]** (2) Pure API

**[0063]** The system provides a standard Notify for terminal software  
5 to report error information; the API is similar to:

```
void Notify (char name[], char provider[], char version[], char  
data[]...);
```

**[0064]** The contents of the arguments may be understood with reference  
to the relevant information in the error description format table.  
10 The approach is easy for use, but poor in extendibility.

**[0065]** When the DM Agent module reports the terminal error information  
through the information reporting interface, it reports the received  
software error information to the DM Server via an extended OMA DM  
protocol. The extended OMA DM protocol may be implemented in any of  
15 the following ways:

**[0066]** 1. Implement with General Alert command of the OMA DM protocol:

**[0067]** The DM protocol supports events triggered by the client with  
General Alert command. The command may be extended to support the  
automatic reporting of terminal error information.

20 **[0068]** The extending of the information description may be made in  
either of the following ways (the fields used hereunder are only  
illustrative; terminal information may be described in a more  
normalized and completed format when the standard is set down; there  
is no restriction in this aspect in the invention):

25 **[0069]** A. Attaching terminal error information description in Data  
field directly:

```
<Alert>  
    <CmdID>1</CmdID>  
    <Data>1226</Data>  
30    <Item>
```

```

5      <Source>
          <LocURI>./SyncML/Sample</LocURI>
      </Source>
      <Meta>
          <Type xmlns="syncml:metinf">
              x-oma-application:syncml.samplealert
          </Type>
          <Format xmlns="syncml:metinf">xml</Format>
          <Mark xmlns="syncml:metinf">critical</Mark>
10     </Meta>
          <Data>//extended portion hereunder
              <Report>
                  <Info></Info>
                  <Level></Level>
15                 <AppName></AppName>
                  <Provider></Provider>
                  <Address></Address>
                  <Version></Version>
                  <DataType></DataType>
20                 <Data></Data>
                  <Date><Date>
                  <OS></OS>
                  <OSVersion></OSVersion>
                  <MemorySize></MemorySize>
25                 </Report>
            </Data>
        </Item>
    </Alert>
```

**[0070]** By adding a tag such as <Report> tag to extend the Data tag,  
30 an automatic error reporting message may be transmitted. When the  
Server receives the message and finds the Report tag, it may determine

that the General Alert is an error reporting message. And, the Server may retrieve response data and trigger a processing flow.

**[0071]** B. Attaching terminal error information in the form of systematic [!CDATA] in Data field:

5 <![CDATA[text content]]>

Attach description information in XML format in CDATA, for example:

<Data>

<![CDATA[

<Report>

10 <Info></Info>

<Level></Level>

<AppName></AppName>

<Provider></Provider>

<Address></Address>

15 <Version></Version>

<DataType></DataType>

<Data></Data>

<Date><Date>

<OS></OS>

20 <OSVersion></OSVersion>

<MemorySize></MemorySize>

</Report>

]]>

</Data>

25 **[0072]** (2) Implement by adding new commands

**[0073]** If the General Alert command is not used, the operation may be accomplished by extending special terminal information reporting commands (e.g., adding a Notify command). The Item tag may be used to contain the error information, the format of which may conform to the terminal information description format.

```
<Notify>
  <CmdID>1</CmdID>
  <Item>
    <Report>
      <Info></Info>
      <Level></Level>
      <AppName></AppName>
      <Provider></Provider>
      <Address></Address>
      <Version></Version>
      <DataType></DataType>
      <Data></Data>
      <Date><Date>
      <OS></OS>
      <OSVersion></OSVersion>
      <MemorySize></MemorySize>
    </Report>
  </Item>
</Notify>
```

20 **[0074]** Detailed software information may be involved in the Item field of the Notify command, so as to assist the developer for debugging and handle them timely.

**[0075]** (3) Implement by extending the Alert command, adding an information reporting Alert command.

25 **[0100]** The standard Alert command for DM may be extended into a terminal information reporting Alert command. The terminal information reporting Alert command is allowed to have the same active reporting capability as the General Alert.

**[0101]** For example, the description field of the General Alert is  
30 described with <Data>1226</Data>. The Data field of the Alert command,

for example, may be extended to 1227.

```
<Alert>  
  <CmdID>1</CmdID>  
  <Data>1227</Data>
```

5                   ...

```
</Alert>
```

[0102] The Alert command may be presented as Alert. The detailed information description format may be determined with reference to the error reporting format for terminal information reporting with the General Alert.

[0103] Referring to Fig.2, the automatic reporting process of terminal device software error is as follows:

[0104] Step 1: an error occurs in the terminal device software; and the information of the error is described in the XML format;

15 [0105] Step 2: the terminal software reports the error information to the DM Agent module through the standard information reporting interface;

[0106] Step 3: the DM Agent module reports the error information to the DM Server via the extended OMA DM protocol;

20 [0107] Step 4: the DM Server forwards the error information to the MU of the terminal software developer through the information reporting interface, so that the MU may perform recording and statistics for software errors. Before reporting the error information, the DM Server may perform addition or deletion on data and format conversion for the error information (e.g., filtering off the user's mobile telephone number and transmitting to the MU).

25 [0108] The DM system may maintain a list of applications and the corresponding MU addresses (or ISP addresses), so that the DM Server may obtain the MU address (ISP address) from the list according to the application indicated in the terminal information. For example,

the addresses list may be stored in the DM Server locally.

**[0109]** Referring to Fig.3. The automatic process of terminal device error reporting and handling is as follows:

**[0110]** Step 10: an error occurs in the terminal device software; and  
5 the information of the error is described in the XML format;

**[0111]** Step 11: the terminal software reports the error information to the DM Agent module through the standard information reporting interface;

**[0112]** Step 12: the DM Agent module reports the error information  
10 to the DM Server via the extended OMA DM protocol;

**[0113]** Step 13: the DM Server searches the local maintenance database to see whether there is corresponding solution information according to the application information indicated in the error information; if there is a matched solution, the DM Server determines the error  
15 may be solved automatically, and the flow proceeds to Step 16; otherwise the flow proceeds to Step 14;

**[0114]** Step 14: the DM Server reports the error information to the terminal software provider's MU through the standard information reporting interface;

**[0115]** Step 15: the terminal software developer analyzes the error,  
20 provides a software package to solve the error, and issues the software package to the DM Server through the information reporting interface;

**[0116]** Step 16: the DM Server implements the terminal software  
25 resolution following the standard OMA DM process.

**[0117]** An embodiment of the terminal device error information reporting and the terminal maintaining is shown as follows:

**[0118]** Suppose the terminal information reported from the ME of user A is:

30 <Report>

```
<Info>Version Error</Info>
<Level>High</Level>
<AppName>TestApp</AppName>
<Provider>Huawei</Provider>
<Address>www.huawei.com</Address>
<Version>1.0.1</Version>
<DataType>text/plain</DataType>
<Data>8001</Data>
<Date>20050505201010</Date>
<OS>MobileOS</OS>
<OSVersion>1.0.2</OSVersion>
<MemorySize>200</MemorySize>
```

</Report>

**[0119]** The terminal information indicates an error coded as 8001 occurs in the program with the application named as TestApp and the version number as 1.0.1.

**[0120]** Suppose the automatic processing information for the local error handling is maintained in the DM Server:

AppName	Version Number	Error Data	Processing Mode	Data
TestApp	1.0.1	8001	01 (here "01" indicates new version update)	/root/update/testapp/1.0.2/update.dat
...	...	...	...	...

**[0121]** After checking the reported error information, the DM Server searches the table for matching; if finding the corresponding processing data there, the DM Server will update the terminal software according to the processing mode and the corresponding data therein, so as to attain the result of automatic error handling. If

the processing information is not found, the DM Server will report the terminal information to the MU (or ASP).

**[0122]** In the flow shown in Fig.3, the DM Server may also be configured not to handle the error but report the error information directly to the terminal software developer's MU for handling.

**[0123]** As for the hardware error information and process information created by terminal devices, the handling procedures are identical to those for the above error information, except for the tag in the description format, which is slightly different to the tag for the above error information.

**[0124]** The embodiments mentioned above are only used to describe and explain the principle of the invention. It is understood that the embodiments of the invention are not limited to those ones. Those skilled in the art may easily make modifications and alterations, without departing from the spirit and scope of the invention; however, any of such alterations or modifications shall fall into the protected domain of the invention. Therefore, the protected domain of the invention shall only be confined by the claims.